
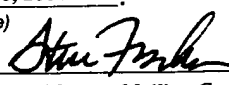


IBM AP

TRANSMITTAL OF APPEAL BRIEF (Large Entity)					Docket No. YOR920000717US1	
In Re Application Of: Hoi Chan et al.						
Application No. 09/916,943	Filing Date July 27, 2001	Examiner Joseph P. Hirl	Customer No. 23389	Group Art Unit 2121	Confirmation No. 8815	
Invention: CONFLICT-HANDLING ASSIMILATOR SERVICE FOR EXCHANGE OF RULES WITH MERGING						
<u>COMMISSIONER FOR PATENTS:</u> Transmitted herewith in triplicate is the Appeal Brief in this application, with respect to the Notice of Appeal filed on The fee for filing this Appeal Brief is: \$340.00 <input type="checkbox"/> A check in the amount of the fee is enclosed. <input type="checkbox"/> The Director has already been authorized to charge fees in this application to a Deposit Account. <input checked="" type="checkbox"/> The Director is hereby authorized to charge any fees which may be required, or credit any overpayment to Deposit Account No. 50-0510/IBM <input type="checkbox"/> Payment by credit card. Form PTO-2038 is attached. WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.						
<div style="text-align: center;"> _____ <i>Signature</i></div> Steven Fischman Registration No. 34,594 SCULLY, SCOTT, MURPHY & PRESSER			<div style="text-align: right;">Dated: December 8, 2004</div> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"><div style="font-size: small;">I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to "Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450" [37 CFR 1.8(a)] on <div style="text-align: center;">December 8, 2004 (Date)</div><div style="text-align: center;"> <i>Signature of Person Mailing Correspondence</i> Steven Fischman Typed or Printed Name of Person Mailing Correspondence</div></div></div>			
cc:						



CERTIFICATE OF MAILING UNDER 37 C.F.R. §1.8(a)

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to: Mail Stop Appeal Brief – Patents, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on December 8, 2004.

Dated: December 8, 2004


Steve Fischman

APPEAL BRIEF

Applicants: Hoi Chan et al.
For: CONFLICT-HANDLING ASSIMILATOR SERVICE
FOR EXCHANGE OF RULES WITH MERGING
Application No.: 09/916,943
Filed: July 27, 2001
Examiner: Joseph P. Hirl
Art Unit: 2121

Steven Fischman
Attorney for Appellant
Registration No. 34,594

SCULLY SCOTT MURPHY & PRESSER
400 Garden City Plaza, Suite 300
Garden City, New York 11530
(516) 742-4343

TABLE OF CONTENTS

	<u>PAGE</u>
INTRODUCTION	1
I. REAL PARTY IN INTEREST	2
II. RELATED APPEALS AND INTERFERENCES	2
III. STATUS OF THE CLAIMS	2
IV. STATUS OF THE AMENDMENTS	2
V. SUMMARY OF CLAIMED SUBJECT MATTER	2
VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL	4
VII. ARGUMENT	4
The rejection of Claims 1-18 under 35 U.S.C. § 102(e) as being anticipated by U.S. Patent Publication No. 2002/0107861 to Clendinning et al. is improper.	4
CONCLUSION	
VIII. CLAIMS APPENDIX	9
IX. EVIDENCE APPENDIX	NONE
X. RELATED PROCEEDINGS APPENDIX	NONE

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES**

Applicant(s):	Hoi Chan et al.	Examiner:	Joseph P. Hirl
Serial No:	09/916,943	Art Unit:	2121
Filed:	July 27, 2001	Docket:	YOR920000717US1 (14033)
For:	CONFLICT-HANDLING ASSIMILATOR SERVICE FOR EXCHANGE OF RULES WITH MERGING	Dated:	December 8, 2004
Confirmation No.:	8815		

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

APPEAL BRIEF

Sir:

INTRODUCTION

Pursuant to the provisions of 35 U.S.C. §134 and 37 C.F.R. §§ 1.191 and 1.192, this paper is submitted as a brief setting forth the authorities and arguments upon which Appellant relies in response to the final rejection of Claims 1-18 in the above-identified patent application in the Office Action dated July 8, 2004.

This brief is being filed authorization to charge the fee of \$340 under 37 C.F.R. §1.17(c) to deposit account 50-0510/IBM. This brief is being filed within the time allowed for reply to the action from which the appeal was taken (37 CFR 1.192(a)). Accordingly, no late fee or request for extension of time is needed.

I. REAL PARTY IN INTEREST

The real party of interest in the above-identified patent application is International Business Machines Corporation (IBM) of Armonk, NY.

II. RELATED APPEALS AND INTERFERENCES

Appellant respectfully submits that no other appeals are known to applicants, the applicants' legal representative, or assignee, that will directly affect or be directly affected by, or have a bearing on, the Board's decision in the pending appeal.

III. STATUS OF THE CLAIMS

Claims 1-18 are pending and appealed. Each of these claims is rejected. Claims 1, 9 and 15 are independent claims.

IV. STATUS OF THE AMENDMENTS

An amendment/response was filed to the Final Office Action dated July 8, 2004, however, this amendment was not entered. Accordingly, there is one un-entered amendment.

V. SUMMARY OF CLAIMED SUBJECT MATTER

The invention relates generally rule based systems, a form of knowledge based systems, that originate from the context of Artificial Intelligence. Particularly, the present invention has grown out of a growing requirement and trend in e-business and network-centric computing of inter-operability, including the use of rule-based systems that are part of heterogeneous applications. The present invention particularly relates to rule-sets applied in rule-based systems, e.g., expert systems that encode knowledge of a human expert, and comprise logic. Logic, in such rule-based systems are represented by rule-sets and comprise rules, e.g., provided in the form of if-then-else patterns (specification page 1, lines 16-20), that are implemented specifically to separate the logic from data which are facts and assertions. Rule-based systems or, expert systems, are implemented in any human knowledge context, for example, business policy, and

lends itself to implementation of rule-based applications. For example, as set forth in the specification at page 8, lines 5-17, and at page 9, lines 11-14, the rule-sets include policy (e.g., a business policy). Thus, for example, one application is a supplier who imports the requirement of a customer, and conducts the negotiation of contract between multiple parties. The sharing and merging of rules for e-business automation involves rules with different formats and will inevitably produce conflicts among rules. An important issue in rule systems is thus conflict handling where two rules within the same ruleset (e.g., program, agent, knowledge base) may lead to incompatible, i.e., mutually exclusive, conclusions. For example, a first rule classifies an incoming message as having a very high degree of urgency, while another rule classifies the message as having a medium degree of urgency (see specification, page 6, lines 9-14). Therefore, a system to handle conflicts is an important part of the updating, editing of existing rules as well as rule-sharing and merging. The lack of a systematic way to resolve conflict is the problem that the present invention is directed to solve.

Thus, the present invention is directed to a system and method for merging two rulesets provided in rule-based systems associated with originating applications, e.g., executing at different locations. Each ruleset comprises rules in potential conflict with each other, and each ruleset is in a different rule format. The rulesets to be merged are communicated to an assimilator service provided with a merge policy comprising a set of specifications of partially-ordered priorities and/or mutual-exclusion constraints. The rulesets are translated into a common representation capable of being implemented in any logic program rule engine provided in a rule-based application at any location. The rulesets are assimilated to produce a new merged ruleset comprising logic required for resolving potential conflicts among rules in accordance with the merge policy that is implemented in any logic program rule engine provided at any location. The new merged ruleset is then translated into one of the originating application's rule format.

Independent Claim 9 sets forth such an assimilator system 10 for merging two or more rulesets (R1, R2, Figures 1 and 2) provided in rule-based systems (S1, S2, respectively, Figures 1 and 2) associated with originating applications (App_1, App_2, respectively, Figures 1 and 2)

executing at different locations, with each ruleset having rules implementing a policy in potential conflict with each other. The system comprises:

a communications network (Fig. 1, network 10, 11; page 11, lines 2 and 3) enabling the transmission and receipt of rulesets (R1, R2, Figures 1 and 2) to be merged between said different locations;

a translator mechanism (Fig. 2, Interlingua 20; page 13, lines 17-20) for translating each ruleset from its rule format into a common core representation (a Courteous Logic Program or “CLP”) capable of being implemented in any logic program rule engine provided in a rule-based application at any location and for translating from the common core representation into each originating application’s rule format;

a conflict transformer mechanism (Fig. 2, conflict transformer sub-system 15; page 14, lines 17-20) for receiving each ruleset and assimilating the rulesets to produce a new merged ruleset (R3, Fig. 2) in accordance with a merge policy (Fig. 2, merge policy 25; page 17, lines 6-20), the new merged ruleset comprising specification of a set of partially-ordered priorities and/or mutual-exclusion constraints that comprise logic required for resolving potential conflicts among rules; and,

a device for translating (Fig. 2, assimilator service 19 and Interlingua package 20; page 16, lines 10-17) the new merged ruleset into a common core representation capable of being implemented in any logic program rule engine provided in a rule-based application at any location.

Independent Claim 1 is a method claim for merging two rulesets (R1, R2, Figures 1 and 2) provided in rule-based systems (S1, S2, respectively, Figures 1 and 2) associated with originating applications (App_1, App_2, respectively, Figures 1 and 2) executing at different locations, each ruleset comprising rules implementing a policy in potential conflict with each other, and each ruleset being in a different rule format that corresponds to the system Claim 9, and includes steps of:

a) communicating said rulesets (R1, R2, Figures 1 and 2) to be merged over a distributed network (network 11, Figure 1) to an assimilator service device (Fig. 2, conflict handling and assimilator service 19, page 10-11, bridging sentence) for receiving each said ruleset;

b) providing a merge policy (Fig. 2, merge policy 25; page 17, lines 6-20) to said assimilator device, said merge policy comprising a set of specifications of partially-ordered priorities and/or mutual-exclusion constraints;

c) translating the rulesets (Fig. 2, Interlingua 20; page 13, lines 17-20) into a common core representation capable of being implemented in any logic program rule engine provided in a rule-based application at any location;

d) assimilating the rulesets (Fig. 2, conflict transformer sub-system 15; page 14, lines 17-20) to produce a new merged ruleset (R3, Fig. 2) comprising logic required for resolving potential conflicts among rules in accordance with said merge policy, where said new merged ruleset is in a common core representation capable of being implemented in any logic program rule engine provided in a rule-based application at any location;

e) translating said new merged ruleset into one of said originating application's said rule format (Fig. 2, assimilator service 19 and Interlingua package 20; page 16, lines 10-17); and

f) communicating said translated new merged ruleset over said distributed network to the one of said originating applications.

Independent Claim 15 is a program storage device claim for performing the same method as in Claim 1. The program storage device may be executed in the network devices 12a,...,12n in the networked system 10 of Fig. 1.

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

Whether the rejection of Claims 1-18 under 35 U.S.C. § 102(e) as being anticipated by U.S. Patent Publication No. 2002/0107861 to Clendinning et al. is improper.

VII. ARGUMENT

The rejection of Claims 1-18 under 35 U.S.C. § 102(e) as being anticipated by U.S. Patent Publication No. 2002/0107861 to Clendinning et al. is improper.

Claims 1 and 15

Claim 1 recites a method for merging two rulesets provided in rule-based systems associated with originating applications executing at different locations, each ruleset comprising rules implementing a policy in potential conflict with each other, and each ruleset being in a different rule format, the method comprising:

- a) communicating said rulesets to be merged over a distributed network to an assimilator service device for receiving each said ruleset;
- b) providing a merge policy to said assimilator device, said merge policy comprising a set of specifications of partially-ordered priorities and/or mutual-exclusion constraints;
- c) translating said rulesets into a common core representation capable of being implemented in any logic program rule engine provided in a rule-based application at any location;
- d) assimilating said rulesets to produce a new merged ruleset comprising logic required for resolving potential conflicts among rules in accordance with said merge policy, where said new merged ruleset is in a common core representation capable of being implemented in any logic program rule engine provided in a rule-based application at any location;
- e) translating said new merged ruleset into one of said originating application's said rule format; and
- f) communicating said translated new merged ruleset over said distributed network to the one of said originating applications.

Claim 15 is a program storage device claim for performing the same method as in Claim 1.

The Examiner in the Final Rejection (Office Action page 3) has alleged that Clendinning anticipates the inventive method because a) Clendinning communicates rule-sets to be merged over a network to an assimilator service and cites Clendinning at ¶¶0046 and ¶¶0072-075]; b) provides a merge policy to the assimilator comprising a set of specification of partially-ordered priorities and/or mutual-exclusion constraints [Clendinning at ¶¶0046]; c) translates rule-sets into sets of common core representation capable of implementation in any rule engine provided in a rule-based application; and d) assimilates the rulesets to merge them according to a merge policy

that provides logic required for resolving potential conflicts among the rules [Clendinning at ¶0046].

The Examiner's assertion that Claims 1 and 15 are anticipated by Clendinning et al. ("Clendinning") is disagreed primarily for the reason that Clendinning are only concerned with a system for collecting and displaying data (information) of a product at a web-site and, a related method for storing a product's various identifiers (e.g., attributes/value pairs) in a database, the plurality of identifiers for the product and, the relationships between the identifiers. The crux of the Examiner's argument is the Examiner equivocation of a "database" as a "ruleset". However, one skilled in the art would recognize that these are not the same:

A "database" is defined as a collection of information organized in such a way that a computer program can quickly select desired pieces of data. In a simple example, a database may be thought of as an electronic filing system. Traditional databases are organized by fields, records, and files with a field being a single piece of information; a record one complete set of fields; and a file is a collection of records. For example, a telephone book is analogous to a file. It contains a list of records, each of which consists of fields: name, address, and telephone number. An alternative concept in database design is known as Hypertext. In a Hypertext database, any object, whether it is a piece of text, a picture, or a film, can be linked to any other object. Hypertext databases are particularly useful for organizing large amounts of disparate information, but they are not designed for numerical analysis. Moreover, the definition of datum (data) is an item of factual information, e.g., derived from measurement or research.

To the contrary, a "ruleset" as claimed in independent Claims 1 and 15 is defined as comprising "rules" (e.g., a collection of rules). Respectfully, the definition of a rule may be stated in a variety of ways, including: 1. a principle or condition that customarily governs behavior; 2. a prescribed guide for conduct or action; 3. a rule can contain logic and infer new information; and 4. directions that define the way an activity is to be conducted. Simply stated, "Rules" contain logic, and can infer new information while "Data" are facts/information.

The mechanism to merge logic, i.e., rulesets, as claimed in the present invention thus involves much more analysis than just "data" which is being merged in Clendinning and which does not contain logic. Respectfully, based on the aforementioned definitions that a ruleset and

database are different entities which have different implementations, expressions, purposes, functions, representation and applications, all of the Examiner's reasons in rejecting the independent Claims 1 and 15 are therefore incorrect, even when given a broad interpretation.

At best, the method described in Clendinning et al. functions to collect and normalize data provided from disparate sources, and organize the data for presentation to users over the Internet. The touchstone of Clendinning et al. is the ability to provide a database of disparate product information including their respective attribute/value pairs as provided by vendors in a form that comprises consistent terminology and ontology to facilitate attribute- or parameter-based database searching. Thus, the reason for Clendinning's "assimilation" is to provide the ability to conduct searches in a database [Clendinning ¶0044]. For example, the Clendinning ¶0044 teaches that any particular data may be normalized so that the same data items (attributes and values) describing an object (product) as may be provided by different vendors may be consistent in the database. Thus, as taught in the context of Clendinning ¶0044, a user search of the database for a laptop computer having a screen size "XGA" (i.e., an attribute/value pair) will be able to retrieve a hit for a data point directed to a computer display resolution of "1024x768", i.e. the data domain and attributes/values have been normalized.

To the contrary, assimilation in the present invention, is the generation of a new merged ruleset comprising logic required for resolving potential conflicts among rules in accordance with a merge policy. Rulesets, and claimed in the present invention, have a separate and distinct meaning in the art, and comprises rules, e.g., provided in the form of if-then-else patterns (see, specification page 1, lines 16-20) representing "logic" in rule-based systems, and that are implemented specifically to separate the logic from "data" which are facts and assertions.

While the Examiner is entitled to a reasonably broad interpretation of the claims, and rightly states so in the Final Rejection (pages 3 and 4), applicants' see absolutely no correlation between a rule-set (as claimed) and a database (taught in Clendinning). As claimed, a rule-set comprises rules implementing a policy in potential conflict with each other, and each ruleset being in a different rule format. The "database" in Clendinning basically comprises of vendor product information, e.g., a product ID, a product domain, and attributes (e.g., CPU, display, memory) [see Figs. 2, 3 of Clendinning]. Rule-sets, claimed in the present invention, have a

separate and distinct meaning in the art as comprising rules representing “logic” in rule-based systems that are implemented specifically to separate the logic from the “data” which are facts and assertions stored in the database. Thus, contrary to the Examiner’s indication, “rulesets” as claimed in rejected independent Claims 1 and 15 are not “databases” as the Examiner broadly interprets and asserts in the Final Rejection (Office Action page 9).

Thus, Clendinning system does not anticipate the claims of the present invention that is concerned with rule-sets or rule-based systems implementing a business policy. The main advantage of using a rule-based system in an application is the clear and clean separation of data and business logic, which allows the creation, modification, and maintenance of rules independent of the underlying implementation mechanism.

Clearly, as Clendinning is not oriented to knowledge-based systems, it is not even remotely suggestive of rule-based system applications and the mechanisms for resolving potentially conflicting rules in merged rule-sets. Thus, with respect to the Examiner’s allegation that Clendinning ¶0046 teaches the claimed recitation of a merge policy comprising a set of specifications of partially-ordered priorities and/or mutual-exclusion constraints, Applicants fail to see how the Clendinning at ¶0046 provides this teaching.

In support of this rejection, the Examiner alleges that Clendinning teaches the term partially-ordered priorities specification of a merge policy by citing a passage in Clendinning ¶0046 beginning with the sentence “...In case of conflict with pre-existing information for the product...” and further, alleges that Clendinning teaches the term mutual-exclusion constraints specification of a merge policy by citing step 1003 in Clendinning ¶0046. However, this whole passage is of a whole other context directed to the assimilation of new product data, i.e., vendor product information, to be input to the database and particularly, the ability to represent the data in the database for presentation in a common format via an interface to facilitate user searching, e.g. normalizing the products attributes (as described in the “XGA” example, *supra*) and updating information about them in the databases [see Clendinning Figure 2]. That is, Clendinning’s “assimilation” really is a product mapping: first performing a comparison of an item (product) to be input against a product map (listing vendor products and corresponding identifiers). If the new product is not found in the list [Clendinning ¶0047], it is added (merged)

with new attribute-value pairs, and thereafter, normalized within the database after consultation with various “alias” lists for attributes, values and domains. If the product is already found in the list, a normalization of the domains, attributes and values is initiated in a product specific manner [Clendinning ¶0048].

This is not the same as a merge policy for merging of rule-sets as claimed in the present invention. In the present invention, it is logic in rules that may be of conflict, as illustrated in the following exemplary sense: Two rules within the same ruleset lead to incompatible, i.e., mutually exclusive, conclusions. For example, a first rule of a rule-set classifies an incoming message as having a very high degree of urgency, while another rule of a rule-set classifies that message as having medium degree of urgency (see specification, page 6, lines 9-14). Thus, the output of the present invention as claimed is a new merged ruleset comprising logic required for resolving potential conflicts among rules in accordance with the merge policy. Moreover, terms used in the claims bear a heavy presumption that they mean what they say and have the ordinary meaning that would be attributed to those words by persons skilled in the relevant art. *Texas Digital Systems Inc. v. Telegenix Inc.*, 308 F.3d 1193, 1201-1202, 64 U.S.P.Q.2d 1812, 1817 (Fed. Cir. 2002).

Thus, an example of a merge policy specification in the present invention rather, includes syntax and semantics as described in the specification, e.g., on page 20, et seq. which provides an example of a merge rule represented as follows:

...a(?X,?Y)<-b(?X,?Y....) AND c(?X,?Y....) with a,b,c representing predicates or facts, ?X,?Y representing variables, and with the right hand side of the rule “a(?X,?Y)” representing the conclusion or consequent and, the left hand side of the rule “b(?X, ?Y)” representing the antecedent...

The specification and implementation of a merge policy (that includes syntax and semantics) for expressing conflict resolution as partially-ordered priorities and/or mutual-exclusion constraints is neither taught nor suggested by Clendinning ¶0046 - ¶0048. While Clendinning ¶0046 cited by the Examiner in his rejection of independent Claims 1 and 15 includes terms such as “assimilate” and “merge”, as in the claims of the present invention, it is of a completely different technology context, and does not render the instant invention anticipated.

Accordingly, it can be seen that Claims 1 and 15 are patentable over Clendinning et al. for at least the above reasons. The rejection of these claims is therefore improper.

Claim 9

Claim 9 recites an assimilator system for merging two or more rulesets provided in rule-based systems associated with originating applications executing at different locations, each ruleset having rules implementing a policy in potential conflict with each other, said system comprising:

- a communications network enabling the transmission and receipt of rulesets to be merged between said different locations;

- a translator mechanism for translating each said ruleset from its rule format into a common core representation capable of being implemented in any logic program rule engine provided in a rule-based application at any location and for translating from said common core representation into each said originating application's rule format;

- a conflict transformer mechanism for receiving each said ruleset and assimilating said rulesets to produce a new merged ruleset in accordance with a merge policy, said new merged ruleset comprising specification of a set of partially-ordered priorities and/or mutual-exclusion constraints that comprise logic required for resolving potential conflicts among rules; and,

- device for translating said new merged ruleset into a common core representation capable of being implemented in any logic program rule engine provided in a rule-based application at any location.

Claim 9 of the present invention is an analogous system for carrying out the method steps of and having similar limitations as Claim 1. For the foregoing reasons, Clendinning is not applicable as it does not teach such a system having separate and distinct mechanisms including: a translator mechanism for translating each ruleset from its rule format into a common core representation, nor, a conflict transformer mechanism for receiving each ruleset and assimilating the rulesets to produce a new merged ruleset in accordance with a merge policy, the new merged ruleset comprising specification of a set of partially-ordered priorities and/or

mutual-exclusion constraints that comprise logic required for resolving potential conflicts among rules.

Having already established the major differences between Clendinning and the present invention Claims 1 and 15 *supra*, Claim 9 is patentable over Clendinning et al. for at least the above reasons. The rejection of this claim is therefore improper.

Claim 6

Claim 6, dependent upon Claim 2 and indirectly dependent upon Claim 1, sets forth that the assimilating step includes applying one or more logic mechanisms in the merge policy including mutual exclusion constraints for identifying conflicts and resolving conflicts among the rules.

The Examiner has rejected this claim (Office Action, page 5) by alleging that Clendinning ¶0046 anticipates a logic mechanism including mutual exclusion restraints. The Examiner specifically asserts that what is meant by mutual exclusion is “independence” and that an integer is an example of independent constraint. This is a very simplistic an interpretation and reflects a lack of consideration that “data” are different than “rules” which contain logic. Thus, applicants submit that representation of mutual exclusion by an integer with logical rules is impossible. For example: the mutual exclusion expression: $a(?X1, ?Y2)$ and $b(?X2, ?Y2)$ cannot exist simultaneously unless $?X1 = ?X2$ AND $?Y1 \neq ?Y2$ where $?X1, ?X2, ?Y1$ and $?Y2$ are variables.

The ruleset merging of the present invention uses the above mutual exclusion expression in resolving conflict. Thus, applicants’ do not understand how Clendinning's integer approach can be applied to such a system in resolving conflict since $?X1, ?X2, ?Y1, ?Y2$ are not static values (but “data” are). Thus, the Examiner’s assertion that an integer is an independent entity is incorrect, and the rejection of Claim 6 is erroneous.

Claim 17

Claim 17, dependent upon Claim 15, sets forth that the step of transforming the new merged ruleset from the common core representation back to an originating format after the assimilating step.

The Examiner has rejected this claim (Office Action, page 7) by alleging that Clendinning's teaching of transforming sets of data into common format data anticipates the step of translating rulesets into a common representation. However, the data transformation taught in Clendinning involves canonical representation only. However, in the present invention, the transformation of rulesets into a common representation involves both semantic and canonical representation. In other words, the translated ruleset (into the common representation) must be LOGICALLY equivalent to the original ruleset. The Examiner has failed to consider such point which constitutes and forms the basis of the ruleset assimilation service according to the present invention.

The rejection of Claim 17 is therefore improper.

CONCLUSION

In view of the above, the reference applied against Claims 1-18 does not anticipate these claims under 35 U.S.C. §102(e). Accordingly, Applicants respectfully submit that the rejection is in error and must be reversed.

The Commissioner is hereby authorized to charge any additional fees or credit any overpayment in connection herewith to Deposit Account No. 50-0510/IBM.

Respectfully submitted,

A handwritten signature in black ink, appearing to read "Steven Fischman", with a long horizontal flourish extending to the right.

Steven Fischman
Reg. No. 34,594

SCULLY SCOTT MURPHY & PRESSER
400 Garden City Plaza, Suite 300
Garden City, New York 11530
(516) 742-4343

SF:gc

VIII. CLAIMS APPENDIX

CLAIMS ON APPEAL: Claims 1-18

Application Serial No. 09/916,943

Claim 1: A method for merging two rulesets provided in rule-based systems associated with originating applications executing at different locations, each ruleset comprising rules implementing a policy in potential conflict with each other, and each ruleset being in a different rule format, said method comprising:

a) communicating said rulesets to be merged over a distributed network to an assimilator service device for receiving each said ruleset;

b) providing a merge policy to said assimilator device, said merge policy comprising a set of specifications of partially-ordered priorities and/or mutual-exclusion constraints;

c) translating said rulesets into a common core representation capable of being implemented in any logic program rule engine provided in a rule-based application at any location;

d) assimilating said rulesets to produce a new merged ruleset comprising logic required for resolving potential conflicts among rules in accordance with said merge policy, where said new merged ruleset is in a common core representation capable of being implemented in any logic program rule engine provided in a rule-based application at any location;

e) translating said new merged ruleset into one of said originating application's said rule format; and

f) communicating said translated new merged ruleset over said distributed network to the one of said originating applications.

Claim 2: The method according to Claim 1, wherein said assimilator device is employed to merge rulesets in two or more rule formats from two or more originating applications and communicate the translated new merged ruleset to one of said originating applications.

Claim 3: The method according to Claim 1, wherein said assimilator device is employed for updating rules included in a first ruleset imported from a rules-editor device.

Claim 4: The method according to Claim 2, wherein said assimilating step includes applying one or more logic mechanisms in said merge policy for identifying conflicts and resolving conflicts among said rules.

Claim 5: The method according to Claim 2, wherein a logic mechanism includes a priority specification for expressing conflict resolution.

Claim 6: The method according to Claim 2, wherein a logic mechanism includes mutual exclusion constraints.

Claim 7: The method according to claim 2, wherein said core representation includes a courteous logic program.

Claim 8: The method according to Claim 2, wherein said distributed network is the Internet.

Claim 9: An assimilator system for merging two or more rulesets provided in rule-based systems associated with originating applications executing at different locations, each ruleset having rules implementing a policy in potential conflict with each other, said system comprising:

a communications network enabling the transmission and receipt of rulesets to be merged between said different locations;

a translator mechanism for translating each said ruleset from its rule format into a common core representation capable of being implemented in any logic program rule engine provided in a rule-based application at any location and for translating from said common core representation into each said originating application's rule format;

a conflict transformer mechanism for receiving each said ruleset and assimilating said rulesets to produce a new merged ruleset in accordance with a merge policy, said new merged ruleset comprising specification of a set of partially-ordered priorities and/or mutual-exclusion constraints that comprise logic required for resolving potential conflicts among rules; and,

device for translating said new merged ruleset into a common core representation capable of being implemented in any logic program rule engine provided in a rule-based application at any location.

Claim 10: The assimilator system as claimed in Claim 9, wherein said new merged ruleset is produced in said common core representation, said transforming device converting said new merged ruleset into one of said originating formats.

Claim 11: The assimilator system as claimed in Claim 9, wherein said merge policy includes one or more logic mechanisms for identifying and resolving conflicts among said rules.

Claim 12: The assimilator system as claimed in Claim 11, wherein a logic mechanism includes a priority specification for expressing conflict resolution.

Claim 13: The assimilator system as claimed in Claim 12, wherein a logic mechanism includes mutual exclusion constraints for expressing conflict resolution.

Claim 14: The assimilator system as claimed in Claim 9, wherein said communications network includes the Internet.

Claim 15: A program storage device readable by machine, tangibly embodying a program of instructions executable by the machine to perform method steps for merging two rulesets provided in rule-based systems associated with originating applications executing at different locations, each ruleset comprising rules implementing a policy in potential conflict with each other, and each ruleset being in a different rule format, said method comprising:

a) communicating said rulesets to be merged over a distributed network to an assimilator service device for receiving each said ruleset;

b) providing a merge policy to said assimilator device, said merge policy comprising a set of specifications of partially-ordered priorities and/or mutual-exclusion constraints;

c) translating said rulesets into a common core representation capable of being implemented in any logic program rule engine provided in a rule-based application at any location.

d) assimilating said rulesets to produce a new merged ruleset comprising logic required for resolving potential conflicts among rules in accordance with said merge policy, where said new merged ruleset is in a common core representation capable of being implemented in any logic program rule engine provided in a rule-based application at any location;

e) translating said new merged ruleset into one of said originating application's rule format; and

f) communicating said translated new merged ruleset over said distributed network to the one of originating applications.

Claim 16: The program storage device readable by machine as claimed in Claim 15, wherein said assimilator device is employed for updating rules included in a first ruleset imported from a rules-editor device.

Claim 17: The program storage device readable by machine as claimed in Claim 15, wherein after said assimilating step, a step of transforming said new merged ruleset from said common core representation back to an originating format.

Claim 18: The program storage device readable by machine as claimed in 15, wherein said assimilating step includes applying one or more logic mechanisms in said merge policy for identifying conflicts and resolving conflicts among said rules.